MAPS+TPC Code Merge

Merge the TPC development branch into Master #161

Edit

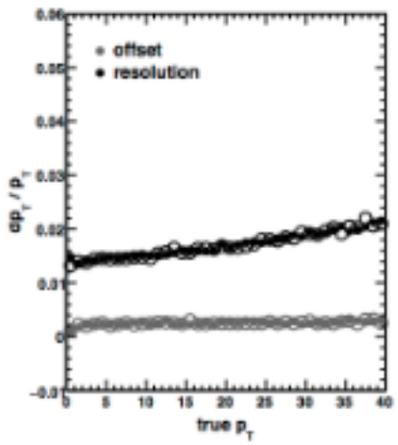
Merged **mccumbermike** merged 54 commits into `sPHENIX-Collaboration:master` from `mccumbermike:tpc_devel_merge` 8 days ago

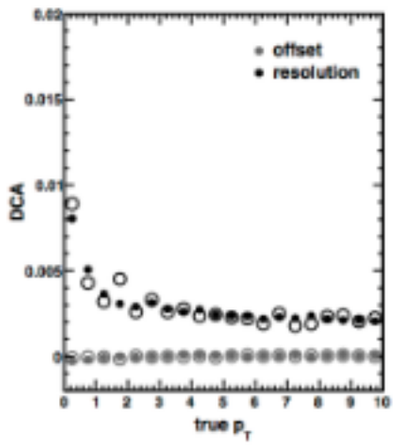
Conversation 3 | Commits 54 | Files changed 37 | +5,821 −484

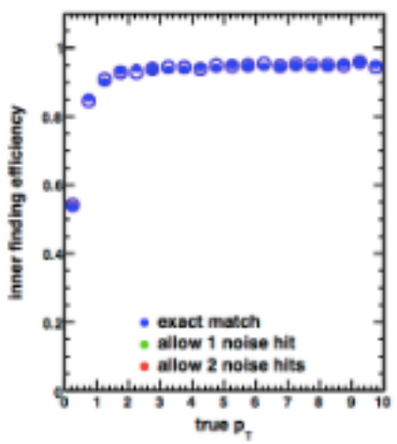
mccumbermike commented 12 days ago — sPHENIX upgrade member

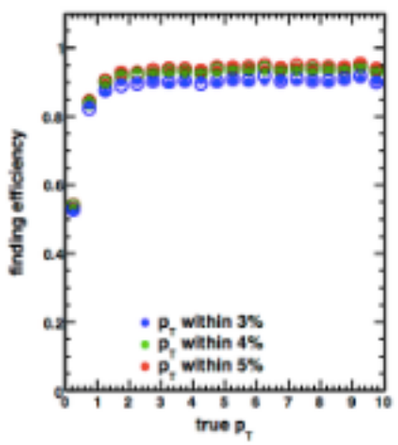
I've forked the TPC tracking code so that it does not interfere with the silicon tracker. Thus with this merge users won't have to switch branches and rebuild to work on the TPC tracking---which will be a focus for the tracking group over the next few weeks.

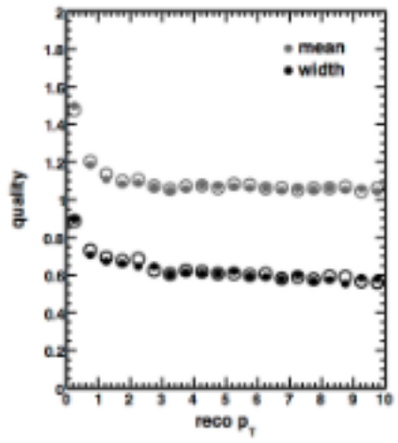
Here is a test before/after for the silicon tracking using a 7-layer MAPS device:

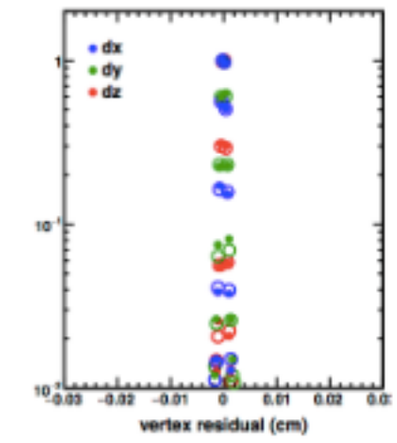


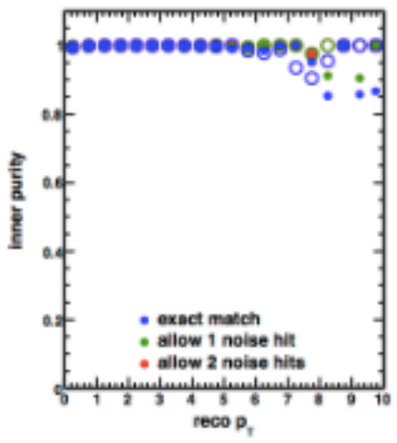


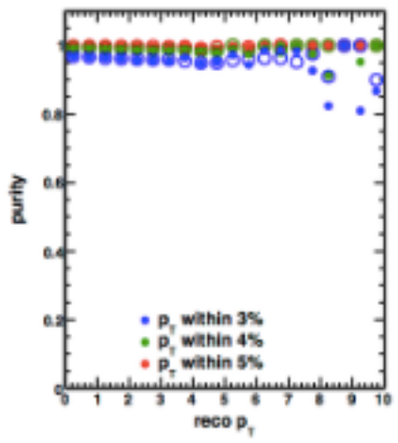












Labels — None yet

Milestone — No milestone

Assignees — No one—assign yourself

4 participants

Notifications

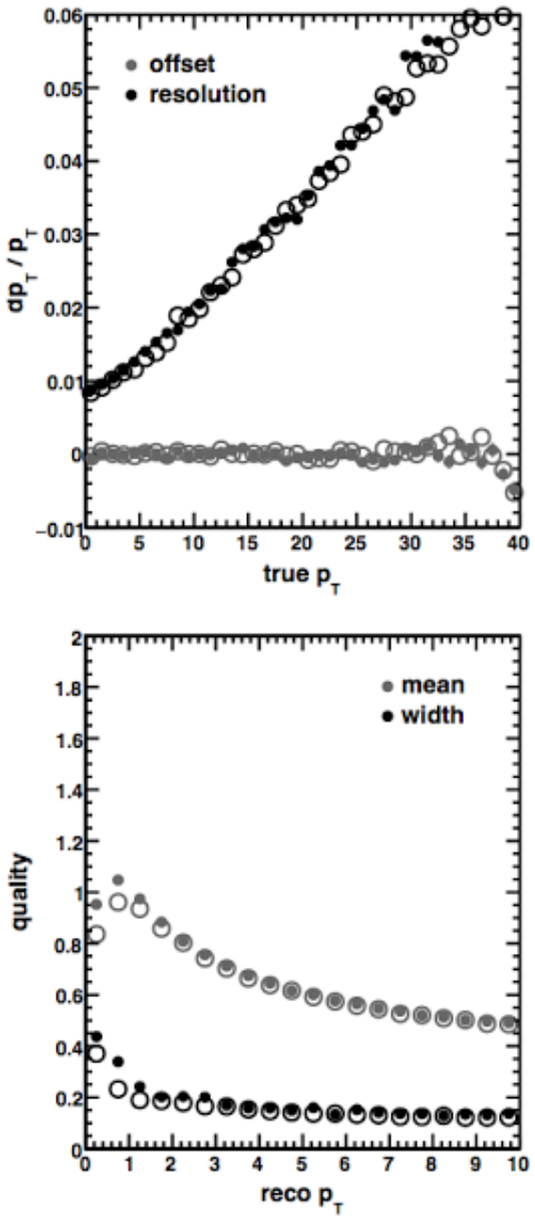
Unsubscribe

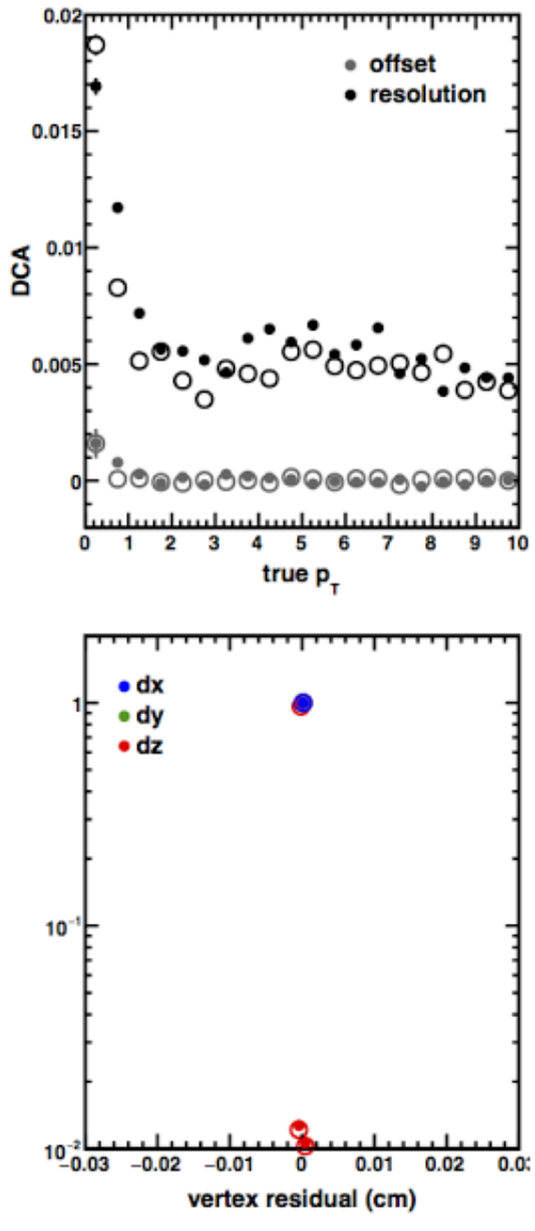
You're receiving notifications because you modified the open/close state.

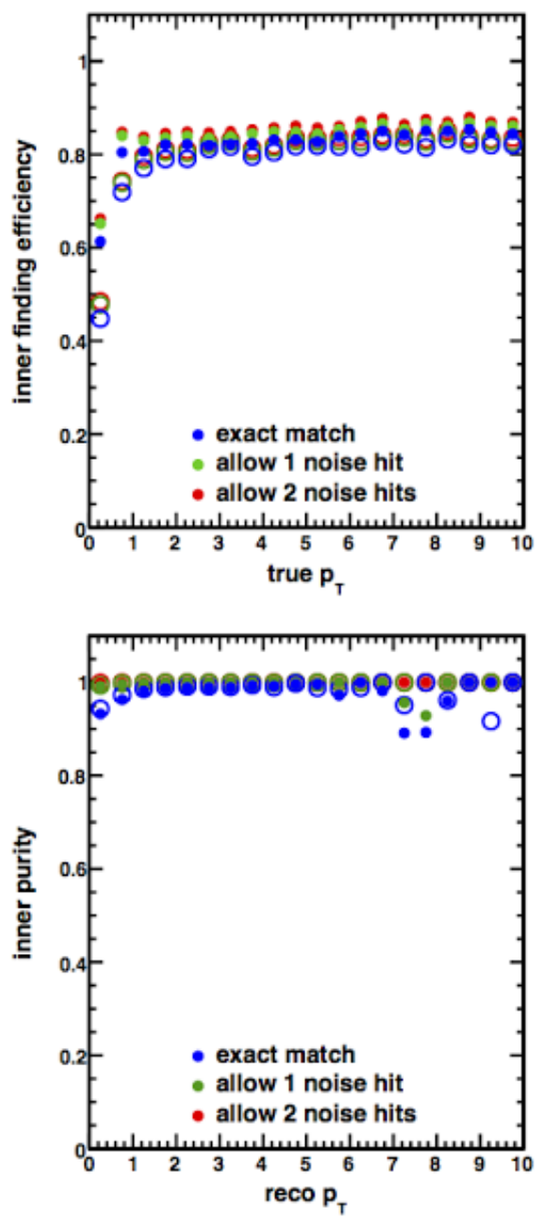
Lock conversation

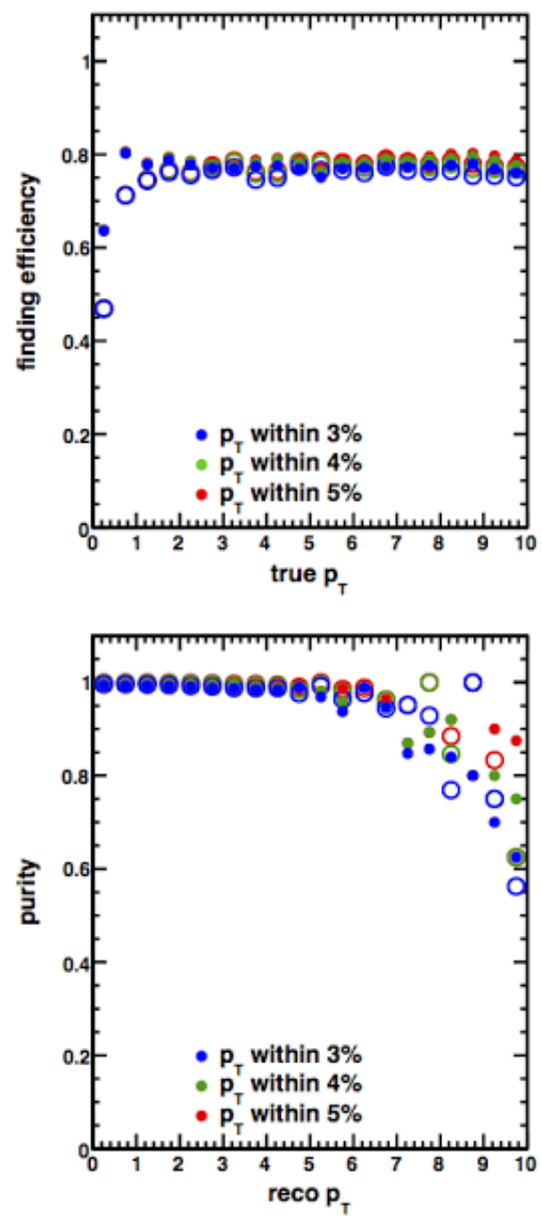
Two Development Branches

Low efficiency, but better fit branch.



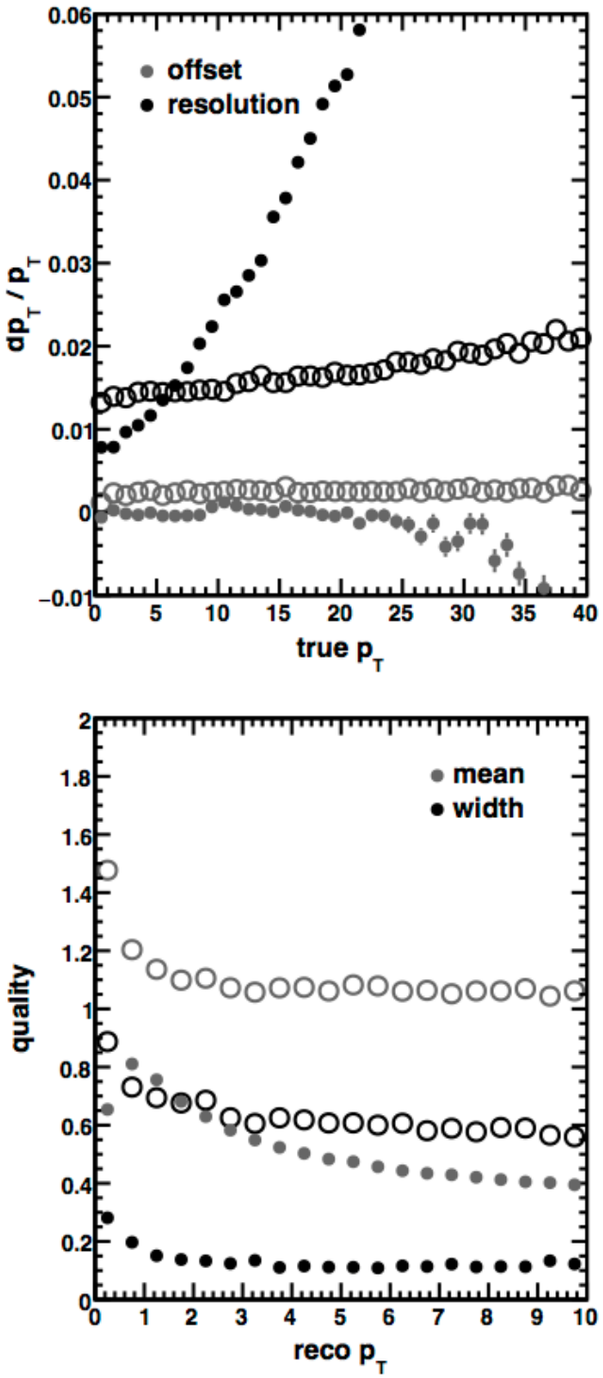


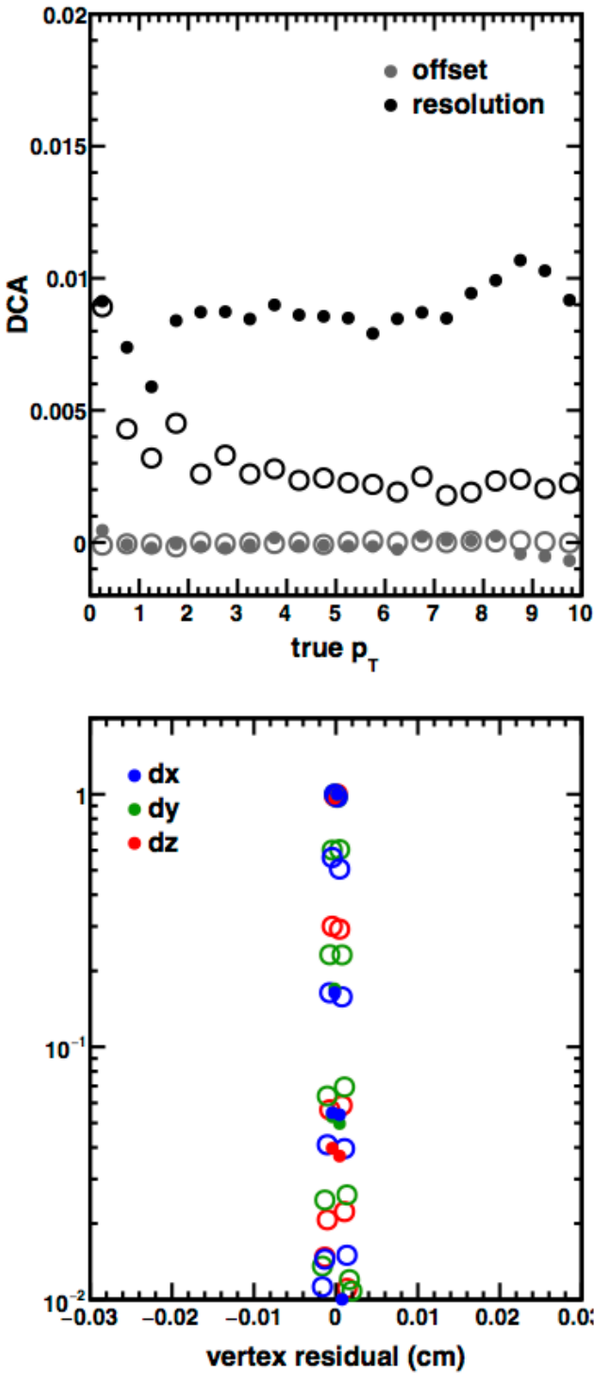


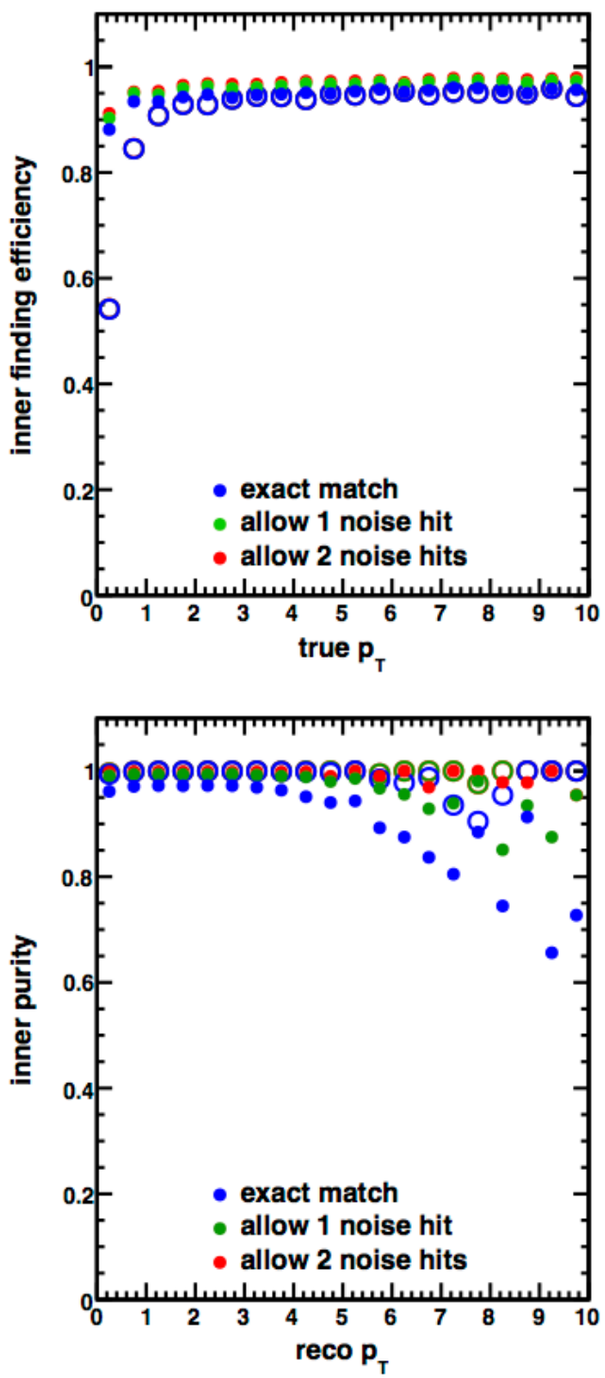


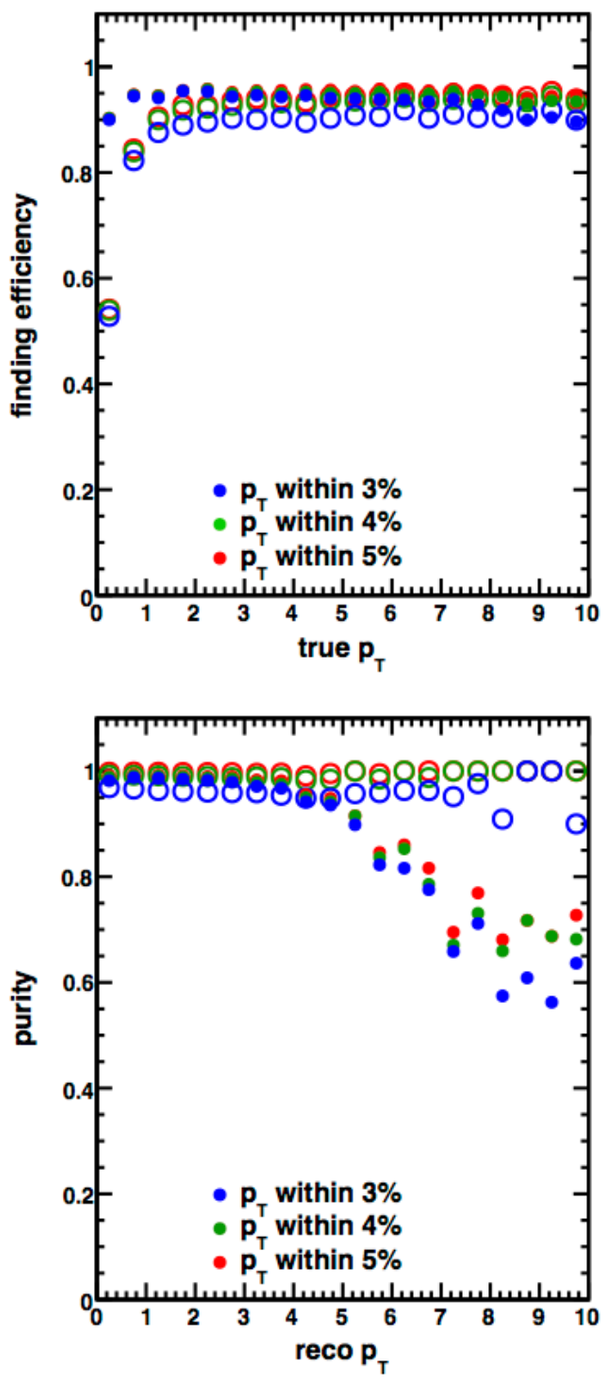
Two Development Branches

High efficiency, but degraded fit branch.







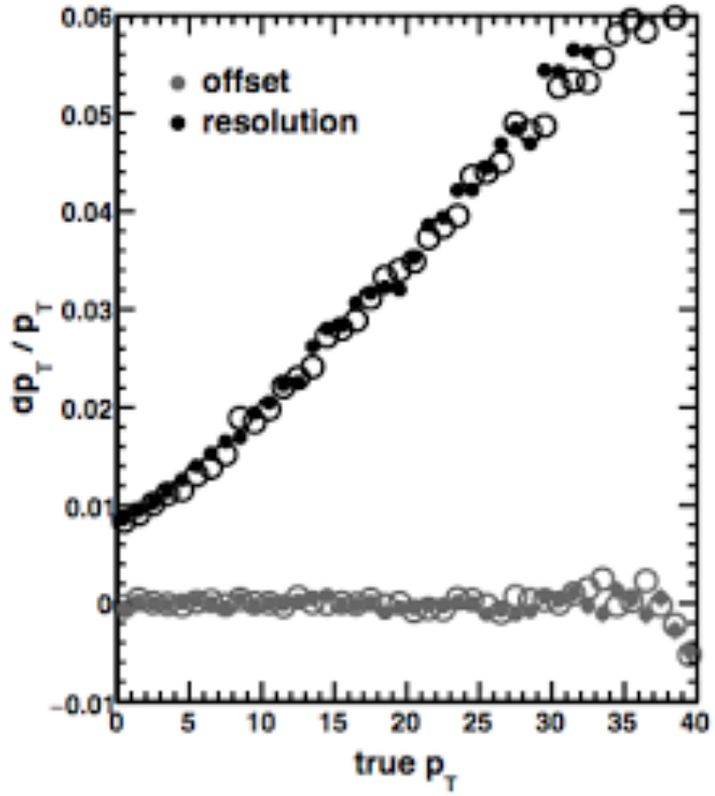


I decided it was probably easier to retune the fit, so merged this one.

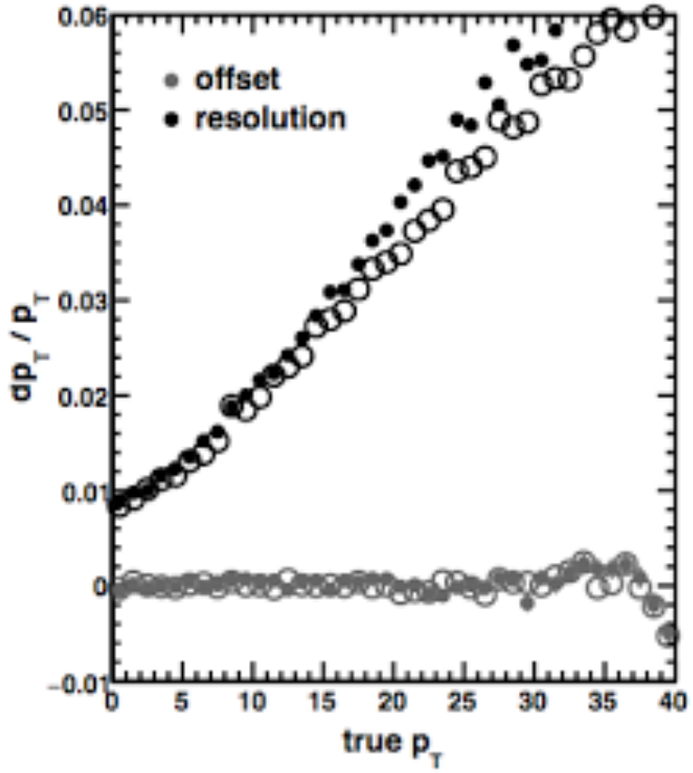
Space Charge

Central HIJING, Low efficiency, better momentum resolution tune

Space Charge (20 cm TPC) - solid points
No Space Charge - open points



Space Charge (30 cm TPC) - solid poin
No Space Charge - open points



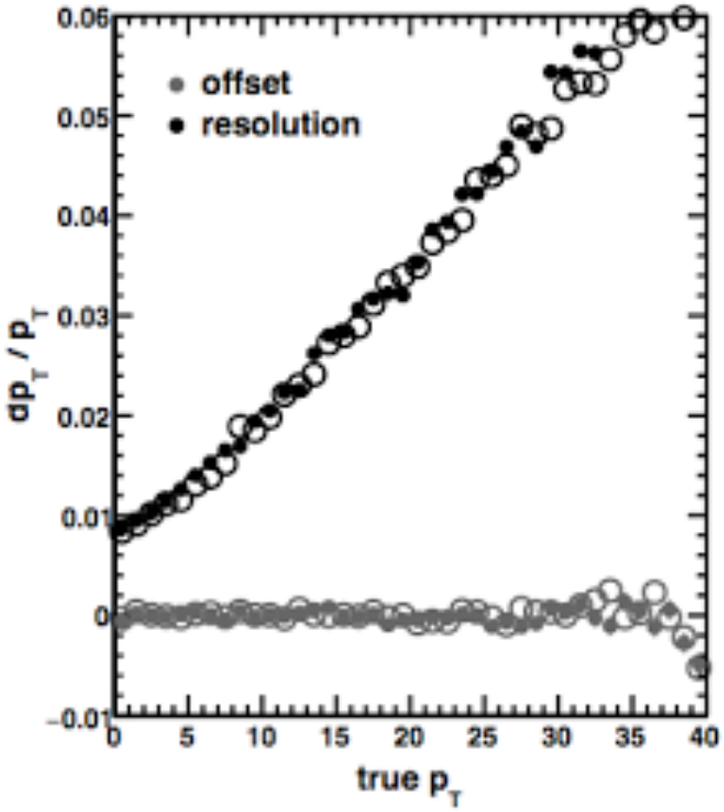
Initial estimates of space charge distortions are very encouraging… only modest degradation at large pt.

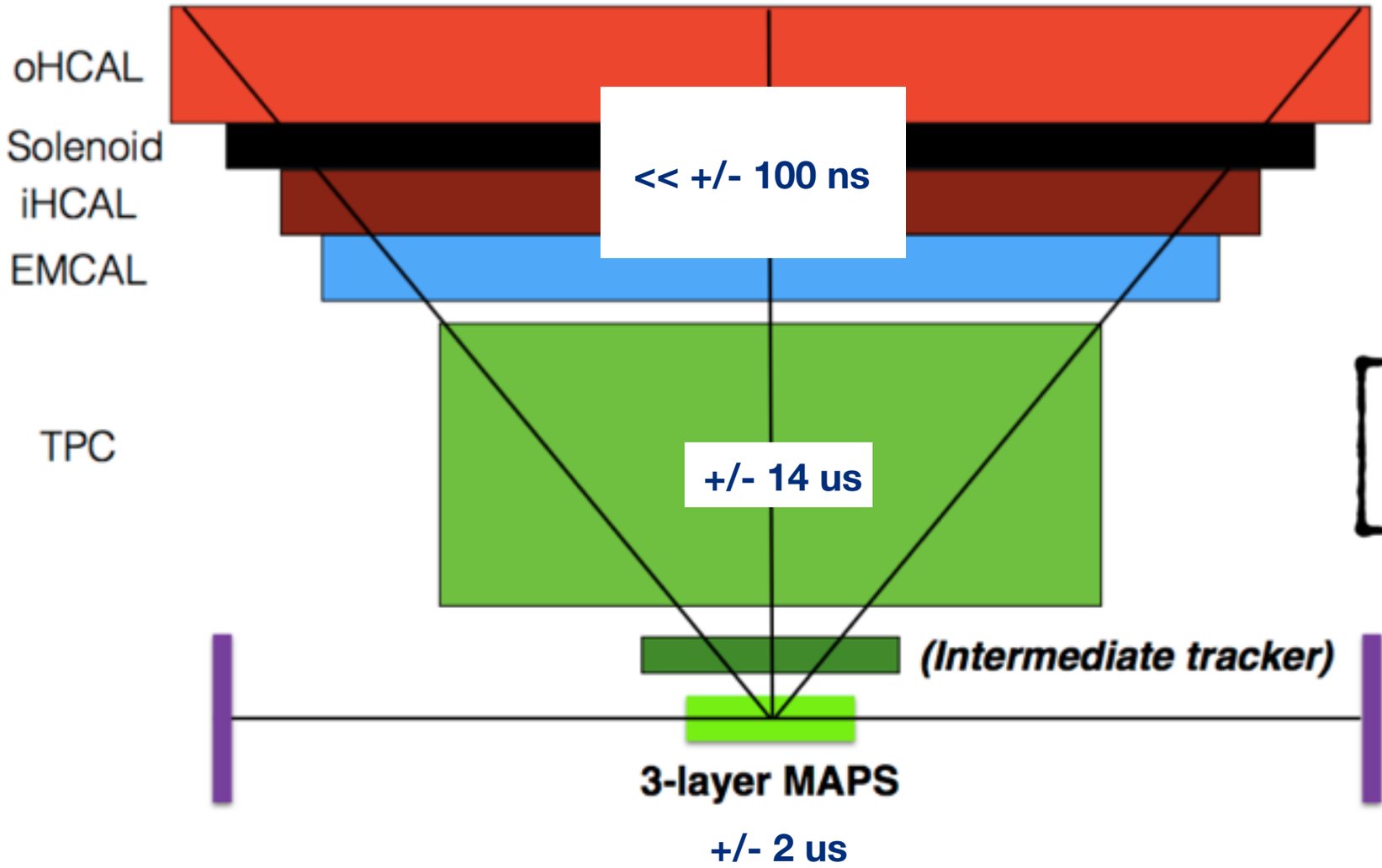
Still some changes to momentum resolution under occupancy to grapple with.

Pileup

I've moved on from thinking **space charge** is the tracking system's biggest open question to thinking about **pileup**.

Space Charge (20 cm TPC) - solid points
No Space Charge - open points





What helps: TPC ~ 6 cm / us drift velocity
Key design parameter: z-pitch of intermediate tracker

Some Next Steps

Event Generator Reorganization:

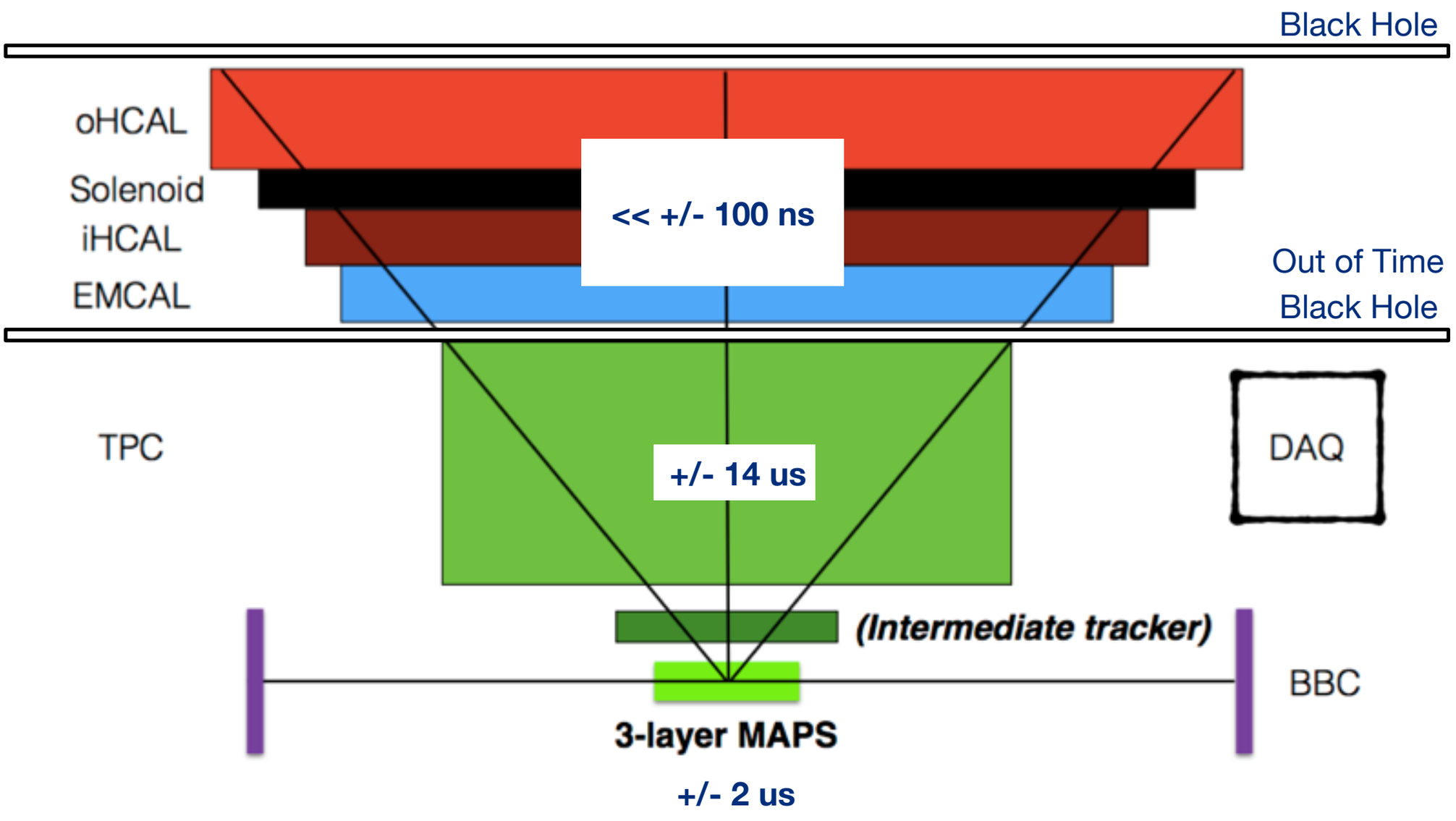
- Multiple entry points (HEPMC & InEvent Node)
- HEPMC interface not re-run safe

~~Layer Dependent Integration Times~~

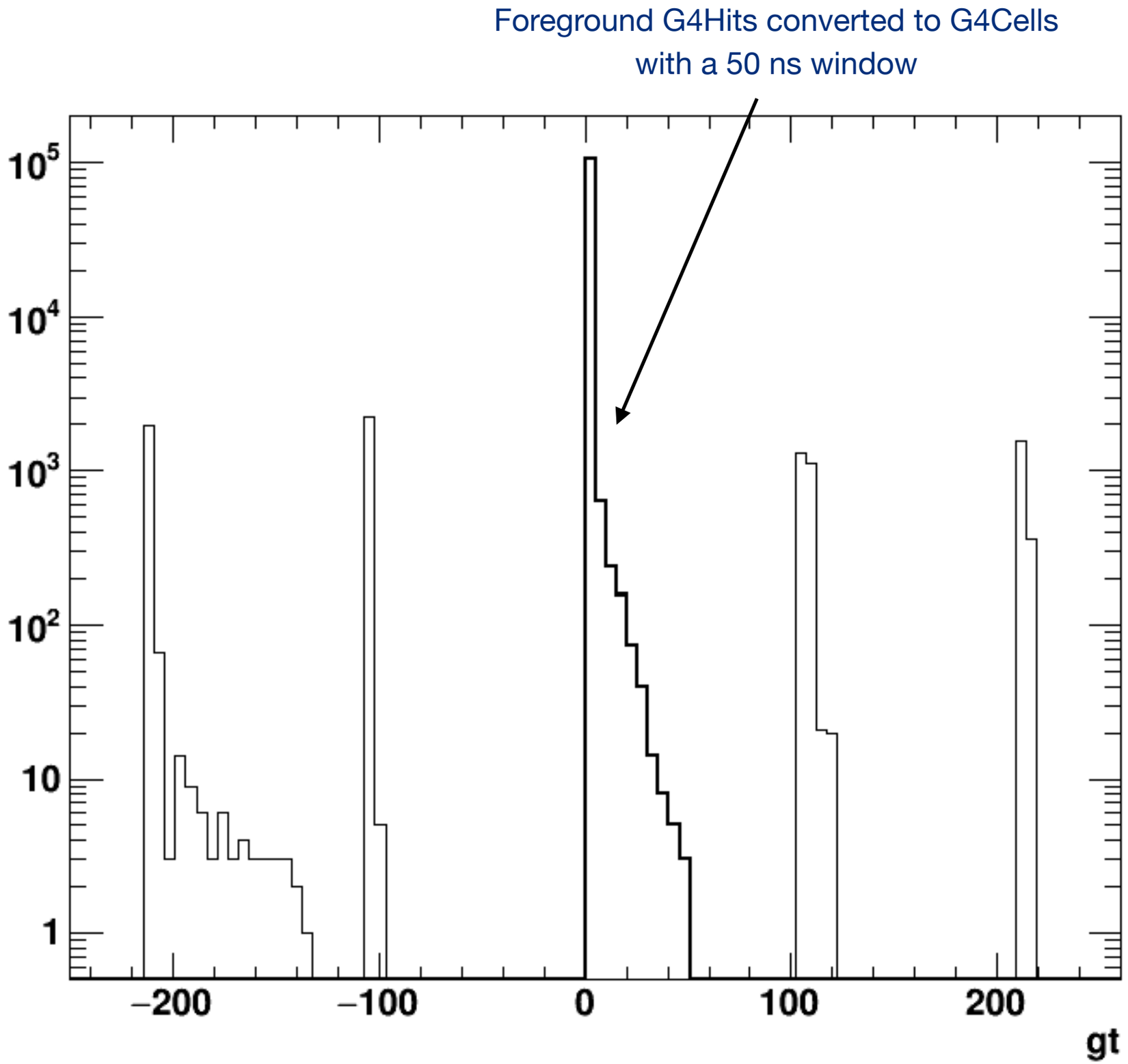
- ~~MAPS & TPC will need different windows~~

~~Prevent CPU usage on out of time hadronic showers~~

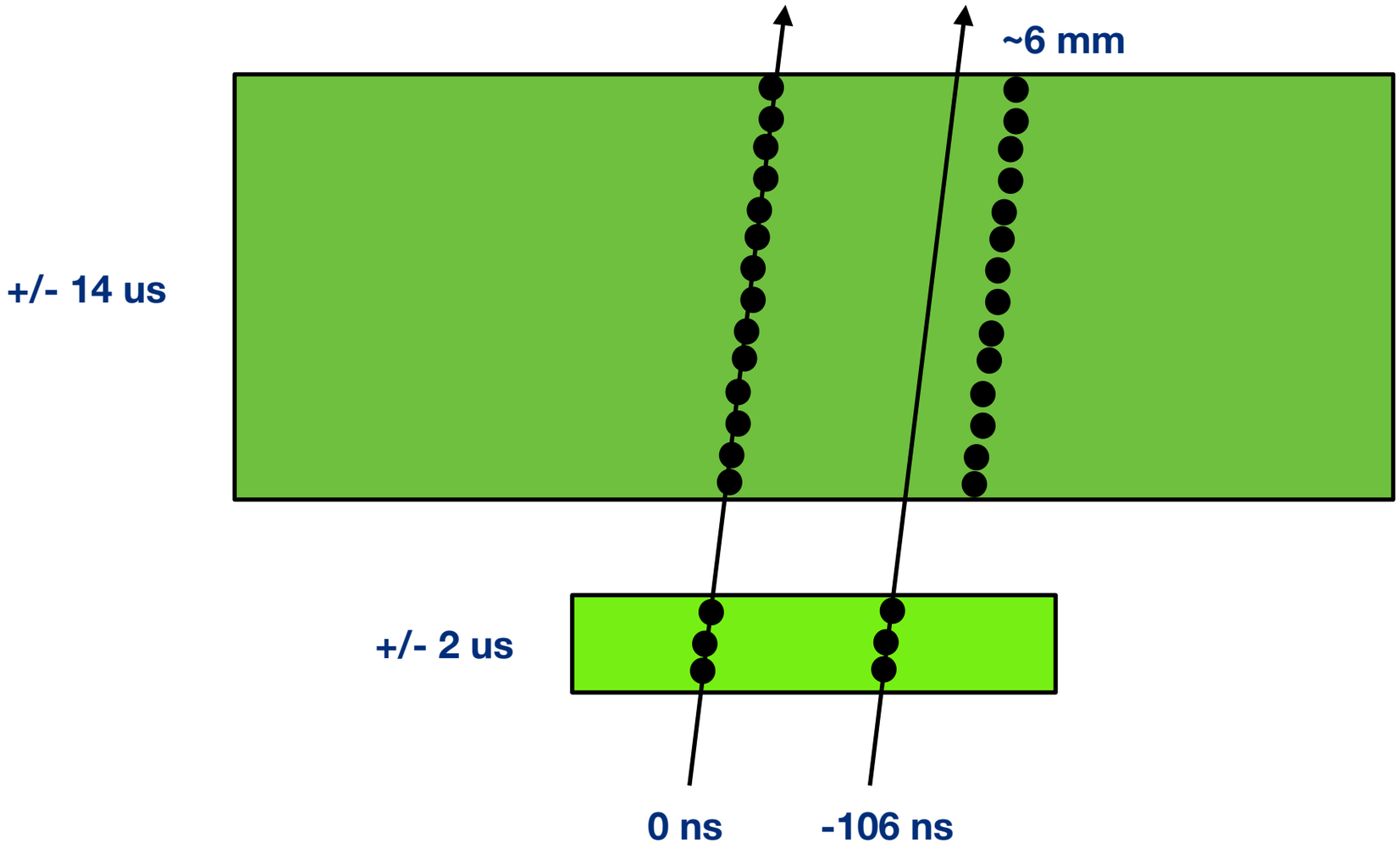
- ~~Time dependent Black Hole~~



Initial Results (Zoomed)



TPC & Out of Time Particles



The drift velocity is rather high, **6 cm/us**, and so out of time tracks will be placed with z-offsets relative to the creation points with **6 mm * crossing offset** which is much larger than the TPC spacial resolution, **O(100 um),** and the MAPS spacial resolution, **28 um**.

I've added this feature to the TPC cell creation, and it **kills the off-time particle fits** after a single crossing offset **via chisq cut**. This should be useful to prevent the reconstruction of full tracks from different crossings.

Of course, occupancy from pileup is still an issue.

TPC Todos

(technical) fix the memory usage in central HIJING (actually from clusterizer?)

=> I will reassess after the meeting the clusterizer usage

```
unsigned int layer = 0;
for(PHG4CylinderCellGeomContainer::ConstIterator layeriter = layerrange.first;layeriter != layerrange.second;++layeriter)
{

        PHG4CylinderCellGeom* geo = geom_container->GetLayerCellGeom(layer);
        nphibins = layeriter->second->get_phibins();
        nzbins =  layeriter->second->get_zbins();

        nhits.clear();nhits.assign( nzbins, 0 );
        amps.clear();amps.assign( nphibins*nzbins, 0. );
        cellids.clear();cellids.assign( nphibins*nzbins, 0 );
```

```
nhits = 942
amps = 3535326
cellids = 3535326
```

(realism) add initial vertexing and remove perfect BBC input from tracking

(performance) improve the track fitting under occupancy, outlier rejection

(technical) improve the passing of uncertainties into HelixHough, remove hard coded errors in TPC version

(performance) remove vertex from fit

Mike: I want to keep pushing on pileup occupancy